

ビデオ オン デマンドシステムにおける 端末上資源管理

1 S-4

羽根 秀宜, 久保 信也, 坂上 秀和, 川崎 成人, 岩崎 未知
NEC C&C研究所

1. はじめに

ネットワーク技術やディスクアレイ技術などの発達に伴い、動画像などの大量のデジタルデータを蓄積したサーバと、動画再生機能などを持つ端末とを高速に結ぶことにより、端末のユーザが必要に応じて動画像の視聴や検索等を行うことのできる、ビデオ オン デマンド(以降VODとする)システムが、マルチメディア産業の基盤の一つとして注目を浴びつつある。

本稿では、まずVODシステムの端末上におけるサービスの構造について述べ、次にアプリケーションプログラムのタスクや実行ファイル、それが使用する各種データファイル等の資源管理について述べる。

2. VODシステムにおけるサービス

この節では、VODシステムの端末上でユーザに提供されるサービスの実現方法とその問題点を述べる。

一般的に、VODシステムでは映画検索やビデオショッピングといったような様々なサービスが、サービスプロバイダ(以下プロバイダと略す)からユーザに対して端末上で提供される。

これらのサービスの実現には端末の構成と兼ね合わせていくつかの方法が考えられるが、ここでは以下のような形態をターゲットとする。

- ・ 端末では、ウィンドウシステムを備えた汎用的なマルチタスクOSが動作している。
- ・ VODの端末として必要な機能をミドルウェアとして備えている。必要な機能とは、サーバとの通信やデジタルビデオデータの再生制御[1]、本稿で述べる資源管理等である。以降、このミドルウェアをVOD Libraryと呼ぶ。
- ・ サービスは、アプリケーションプログラム(以下APと略す)を、サーバから端末上にダウンロードし実行することにより実現する。

このような方法では、サービスを実現するためのAPを各プロバイダが個別に提供することになる。その際、APのモジュラリティが重要となる。これはAP群の独立した開発を容易にすること、個別に開発/提供されたAP群のそれぞれがサービス全体の一部として正しく動作す

ることのためである。

そのためには、次の項目を達成することが必要であると考えられる。

1. 資源の名前空間をサービス単位に分離 あるサービスのAPが実行時に必要とする資源の名前が、他のサービスのAPが必要とする資源の名前と分離され衝突しないこと。
2. 資源の排他 複数のAP間で共有される資源を排他的に利用できること。
3. APの実行管理 あるサービスから別のサービスへの移動は、そのサービスを実現するAPの実行により実現される。そこでAPの実行管理として、まず単純に制御を別のAPへ「遷移」する形態が考えられる。また、サービス全体がツリー構造を持つような場合には、親ノードにあたるサービスではその子にあたる複数のサービスを選択し実行するという状況が考えられる。その際には、親ノードにあたるサービスのAPが子にあたるサービスのAPのひとつを呼び出し、子APが終了するとそれを呼び出した親APに戻るといった、「呼び出し」て戻る形態のAP実行管理を用意することも必要となる。子にあたるサービスが複数のAPから構成されている場合を考えると、呼び出された子APが更に別のAPに「遷移」した際にも、呼び出しを行った親APに戻られるようにする必要がある。

次の節で、上記の1.サービス単位の資源の分離、および3.APの実行管理に焦点を当てた、端末上のVOD Libraryにおける資源管理について述べる。

3. 端末上の資源管理

ここではまずVOD Libraryで管理される資源を定義し、次にそれら資源の管理方法を説明する。

3.1 資源の定義

APは、実行時に静止画など処理を行う上で必要なデータをサーバからダウンロードしこれにアクセスする。また、他のサービスのAPの呼び出し等を行う。そこで、これらに対応した以下の資源を考える。

File: APの実行ファイルやAPが実行時に利用するデータファイル

Task: APの実行主体

次に、これらの管理方法について述べる。

Resource Management On the Set Top Box for Video On Demand System

Hidetaka Hane, Nobuya Kubo, Hidekazu Sakagami,
Shigehito Kawasaki, Michi Iwasaki
C&C Research Laboratories, NEC Corporation

3.2 File管理

Fileの管理で重要となるのは、2節で述べた資源の名前空間のサービス単位での分離である。つまりサービス毎に一意にFileを指定できる必要がある。

そこでサービスを一意に区別するIDであるServiceを導入し、Fileを指定する際に単なるFile名でなくServiceとFile名を組み合わせたFileIDを用いることにより、File名の名前空間をサービス単位に分離する。

図1の例で示すように、Service0におけるFile1は

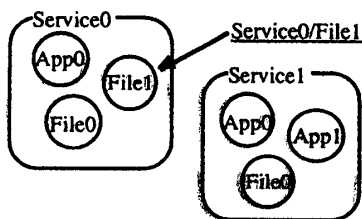


図1: ServiceによるFileの名前空間の分離

Service0/File1というFileIDで指定される。また、File0やApp0といったファイル名がService毎に分離されている。

Serviceの指定方法については、後述のTask管理の節で述べる。

3.3 Task管理

Task管理で重要となるのは、2節で述べたAPの実行管理機能を実現することである。

まず、Taskには一意にTaskを識別するためのTaskIDを割り当てるものとする。

次に、APの実行管理を実現するために、APを起動する際に次の2種類の起動方法を用意する。

Launch: 指定されたAPの「呼び出し」を実現する。これは呼び出し側のTaskは実行を継続しつつ、指定されたAPを別のTaskとして新たに起動する。呼び出し側のTaskは親Task、呼び出された側のTaskは子Taskとなる。子Taskには新しいTaskIDが割り振られる。また親Taskが子Taskの終了を検知するため、子Taskの終了時には親Taskへ終了通知が送信される。

Exec: 指定されたAPへの「遷移」を実現する。これは現在のTaskを終了し、指定されたAPを起動してそのTaskに制御を移す。TaskIDとTask間の親子関係は元のTaskのものが起動されたTaskに継承される。これにより、呼び出されたAPが遷移した場合にも呼び出し元の親Taskへの終了通知の到達が保証される。

また、AP開発時のFileIDの指定を容易にするためにTaskにCurrentServiceという属性を付与する。この値として、それぞれのTaskのAPの実行ファイルのFileIDにおけるServiceの値がセットされる。また、3.2節でのFileIDの指定においてServiceの値が省略された場合は、TaskのCurrentServiceの値が用いられるものとする。したがって

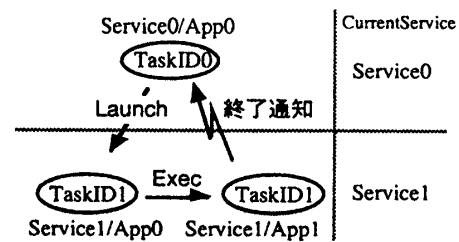


図2: Taskの実行管理の例

そのAPが含まれるService中のFileを指定する際にはServiceの指定を省略することができ、FileIDの指定が容易となる。

これまで述べたTask管理について、図2を用いて具体的に説明する。

1. Service0/App0というFileIDで示されるAPが、TaskIDがTaskID0、CurrentServiceがService0であるTaskとして実行されており、そのAPからService1/App0のAPがLaunchによって呼び出される。この際、LaunchされたAPのTaskには新たにTaskID1というTaskIDが割り振られ、TaskID0のTaskが親、TaskID1のTaskが子となり、CurrentServiceがService1となる。親Taskの実行は継続されるので、バックグラウンドでのFileのダウンロード処理や子Taskの終了待ちの処理などを行うことができる。
2. Service1/App0のTaskからExecによりService1/App1のTaskが新たに起動され制御がそのTaskに遷移する。この際、Service1/App0のTaskからService1/App1のTaskへ、TaskID(TaskID1)とTaskの親子関係が継承される。CurrentServiceの値はService1のままとなる。
3. Service1/App1のTaskが終了すると、Service0/App0のTaskに終了通知が送信される。この通知によりService0/App0のTaskは処理の再開等のアクションを行うことができる。

このような管理方式により、2節で述べた実行管理を実現できる。

4. おわりに

VODシステムにおけるサービスの構造を考慮した端末上の資源管理について述べた。現在、上記の管理方式を持つVOD Libraryを、Microsoft社のWindowsをOSとするPC上で実装している。

今後の課題として、様々なVODサービスに適用し検証を行うこと、大規模Hypermediaアクセス等のサービスでの資源管理モデルの検討などが挙げられる。

参考文献

- [1] 坂上, 羽根, 久保, 川崎, 岩崎 「ビデオ オン デマンド システムにおける分散動画再生制御」, 情報処理学会 第51回全国大会, 1S-03